for example, by time division multiplexing a plurality of broadcast channels, along with other data such as overhead data." See id. at col. 4, lines 56-60. Further, the "file to be transferred via the digital broadcast system such as the system 10 is partitioned at the program center 20 … for transmission as segments 36, as shown in Figs. 3 and 4." See id. at col. 4, lines 48-54. "The segments 36 are provided with headers 37 to facilitate their capture in a local storage device at receiver 14." See id. at col. 4, lines 57-60. "The partitioning of the file 34 allows for smaller portions of the file (e.g., 8kbps channels) to be interspersed with other broadcast content, hereby reducing the demand on the instantaneous bandwidth of the system." Id. at col. 4, lines 54-57. Each segment of the broadcast file includes an associated segment header which is made up of information such as broadcast ID, auxiliary data, file number, segment number, and total segments. See id. Figs. 4-5.

"The receivers are configured to demultiplex a received composite data stream using synchronization symbols and the slot control field data to playback a selected one of the broadcast channels." See id. at col. 3, line 66 – col. 4, line 4. Additionally, receiver 14 functions to capture and store segments. See id. at Fig. 7.

Finally, control center 18 is provided for telemetry, tracking and control of the satellites. See id. at col. 3, lines 51-52.

Independent claim 1

Storing the composite data stream where storing includes: "decomposing ...;" "segmenting ...;" and "discarding ... ."

The system in **Marko** "can broadcast a composite data stream 30 generated, for example, by time division multiplexing a plurality of broadcast channels, along with other data such as overhead data." Id. at col. 3, lines 56-60. In **Marko**, the composite data stream consists of the overhead data and a plurality of time slots for transporting traffic channels. See id. at col. 3, lines 61-65. The overhead data is comprised of a slot control field and master synchronization symbol. Id. Each time slot contains a broadcast segment and its associated segment header which typically includes information such as broadcast ID, auxiliary data, file number, segment number, and total segments. See id. at Fig. 5; col. 4, line 64 – col. 5, line 1. So, the composite data stream in **Marko** comprises all of the following: 1) overhead data including master synchronization symbol and slot control field; 2) a plurality of broadcast segments; and 3) the associated segment headers (broadcast ID, file number, etc ...).

However, neither the **Marko** transmitter nor the **Marko** receiver stores the composite data stream where storing includes: "decomposing ...;" "segmenting ...;" and "discarding ... ." Although the transmitter in **Marko** stores the composite data stream in order to transmit the same, the transmitter does not decompose, segment or discard as a part of its storing. Concomitantly, the receiver in **Marko** does not store the composite data stream. The receiver only stores the segments and does not store either the overhead data or the associated segment headers. See id. at col. 6, lines 41-45 ("[a]s shown in FIG. 7, the converter 55 comprises an RF-to-audio converter 54 and an RF-to-control data converter 56 to extract, respectively, the traffic (e.g. the segments 36) and control data (e.g., headers 37) from the received signal"); see also id. at col. 6, lines 51-55 ("[i]n accordance with the present invention, the converter 55 removes the segment headers 37 from the received data stream and determines from the broadcast ID field 38 or the file number field 40 whether the segment is intended for that receiver 14"). The transmitted segment headers are not stored in the **Marko** receiver's local memory. The only function of the segment headers is to facilitate the "capture and storage of segments." See id. at col. 6, lines 8-9.

Additionally, the receiver in **Marko** only stores **one of many** of the segments which are broadcast. The **Marko** receiver only stores those parts of the broadcast (composite) data stream that the **user selects** at the receiver. See id. at col. 2, lines 58-62 ("the receiver is configured to monitor the codes corresponding to the content in the broadcast signal and to capture content segments having one of the codes selected by the user"). The segment header information is compared with the user-selected broadcast ID or file number to determine which segments to store in local memory. See id. at col. 6, lines 58-59 ("[i]f the segment is intended for that receiver, it is stored in the local storage device 50"). Thus, it is only the **user-selected** segments that are stored in local memory and not any other of the broadcast segments. Even if **Marko** could be redesigned to store the entire composite data stream (which consists of the overhead data, the plurality of broadcast segments, and the plurality of associated segment headers) in the receiver's local memory, it still only stores the **one of many** segments of the broadcast data stream selected by the user. One of many is not the total broadcast transmitted, and is not the composite data stream. Since the overhead data, the de-selected plurality of broadcast segments, and the segment headers associated with the plurality of de-selected broadcast segments are not stored in local memory of the receiver in **Marko**, the entirety of the composite data stream is not stored. For this reason, the **Marko** reference does not anticipate claim 1 because neither the **Marko** transmitter nor the **Marko** receiver stores the **composite data stream** where storing includes: "decomposing ...;" "segmenting ...;" and "discarding ... ."

Decomposing composite data stream and segmenting its constituents

The Office Action states that **Marko** teaches "[d]ecomposing the composite data stream into a plurality of constituent data streams," and "segmenting at least one of the plurality of constituent data streams." [Office Action, p. 3].

However, the **Marko** reference does not disclose segmenting the constituent data streams. In **Marko**, once the composite data stream is received at the receiver, it is converted "to extract, respectively, the traffic (e.g., the segments 36) and control data (e.g., headers) from the received signal." See id. at col. 6, lines 40-45. The system disclosed in **Marko** does not perform segmentation on the resulting segments. Instead, the system in **Marko** stores the segments as they were received. See id. at col. 6, lines 58-59 ("[i]f the segment is intended for that receiver, it is stored in the local storage device 50").

In contrast, claim 1 requires decomposing the composite data stream into a plurality of constituent data streams prior to the segmenting operation. Thus, under claim 1, the composite data stream is first decomposed into its constituent data streams, and, only then, is segmentation performed. This is done to increase compression efficiency, and reduce storage consumption. See spec. at p. 18. For example, independent claim 1 recites as currently amended:

1. A computer implemented method for storing data comprising:

storing a composite data stream so that it may be restored, said storing including,

decomposing the composite data stream into a plurality of constituent data streams;

segmenting at least one of the plurality of constituent data streams decomposed from the composite data stream; and

discarding those of the segments resulting from said segmenting which are determined to have been stored previously.

Since the result of decomposing the composite data stream is "a plurality of constituent data streams," and segmenting is performed on constituent data streams; it necessarily follows that a particular composite data stream must be decomposed into its constituent data streams prior to segmenting its constituent data streams. This is performed in this order to facilitate compression efficiency, and reduction of storage consumption. Thus, the amended language, "constituent data streams decomposed from the composite data stream" is non-narrowing because the limitation "decomposed from the composite data stream" was inherent and already present in claim 1.

Therefore, since there is no segmentation performed on the traffic data (segments) in **Marko**, and the traffic data is a constituent of the composite data streams; there necessarily is no segmentation of the constituent data streams as required by claim 1. Thus, **Marko** does not disclose "segmenting at least one of a plurality of constituent data streams" as required by the claim 1.

Accordingly, Applicant submits that **Marko** does not anticipate claim 1 and respectfully requests withdrawal of the claim rejection.

Dependent claims 3 and 6

Claims 3 and 6 are each dependent on claim 1 and incorporate all the limitations contained therein. Therefore, for at least the same reasons as claims 1 is not anticipated by **Marko**, Applicant submits claims 3 and 6 are also not anticipated. Accordingly, Applicant respectfully requests the withdrawal of the claim rejections.